1. Весы

Напишите класс **Balance** для описания весов с двумя чашами. На левую и правую чашу объекта будут добавляться грузы с различным весом, ваша задача определить положение чаш.

Метод add\_right принимает целое число — вес, положенный на правую чашу весов, add\_left — на левую чашу. Метод result должен возвращать символ =, если вес на чашах одинаковый, R — если перевесила правая, L — если перевесила левая.

Пример 1

|  |  |
| --- | --- |
| Ввод | Вывод |
| balance = Balance()  balance.add\_right(10)  balance.add\_left(9)  balance.add\_left(2)  print(balance.result()) | L |

Пример 2

|  |  |
| --- | --- |
| Ввод | Вывод |
| balance = Balance()  balance.add\_right(10)  balance.add\_left(5)  balance.add\_left(5)  print(balance.result())  balance.add\_left(1)  print(balance.result()) | =  L |

1. Вектор на плоскости

Напишите класс MyVector, описывающий геометрический вектор и операции с ним.

Операции: сложение, вычитание, умножение на константу (в случае, когда константа стоит слева и справа, умножение \*=), вывод , сравнение на равенство и неравенство, длина вектора. Данные операции необходимо реализовать через **специальные функции**.

Пример 1

|  |  |
| --- | --- |
| Ввод | Вывод |
| v1 = MyVector(-2, 5)  v2 = MyVector(3, -4)  v\_sum = v1 + v2  print(v\_sum) | MyVector(1, 1) |

1. Minecraft

В игре Minecraft все игровые объекты делятся на три типа: блоки, сущности и предметы.  
Блоки – это «кирпичики», из которых состоит весь игровой мир.  
Сущности – это все динамические и движущиеся игровые объекты.  
Предметы – это все остальные объекты.

Реализуйте иерархию классов для игры Minecraft:

* BaseObject – базовый класс для всех объектов. Интерфейс этого класса состоит из двух методов:
  1. \_\_init\_\_(x, y, z) – конструктор класса, который принимает на вход 3 числа (координаты объекта в игровом мире)
  2. get\_coordinates() – метод, возвращающий список координат x, y, z (именно в таком порядке)
* Block – класс, наследующийся от BaseObject. У этого класса есть дополнительный метод shatter(), который ничего не принимает на вход и ничего не возвращает. При вызове этого метода все три координаты объекта Block должны стать равны None.
* Entity – класс, наследующийся от BaseObject. У этого класса есть дополнительный метод move(x, y, z), который принимает на вход 3 числа (новые координаты объекта) и ничего не возвращает. При вызове этого метода координаты объекта должны стать равны x, y, z соответственно.
* Thing – класс, наследующийся от BaseObject. У этого класса нет дополнительных методов.

1. Сложение многочленов

Реализуйте класс Polynomial для вычисления многочлена при заданном х, а также для сложения многочленов.

Конструктор класса Polynomial(coefficients) принимает список из коэффициентов.

Вызов экземпляра класса (необходимо использовать соответствующий специальный метод) вычисляет значение в переданной точке.

В результате сложения двух многочленов долже создаваться новый экземпляр класса Polynomial.

**Пример 1**

|  |  |
| --- | --- |
| Ввод | Вывод |
| poly = Polynomial([10, -1])  print(poly(0))  print(poly(1))  print(poly(2)) | 10  9  8 |

### Пример 2

|  |  |
| --- | --- |
| Ввод | Вывод |
| poly1 = Polynomial([0, 1])  poly2 = Polynomial([10])  poly3 = poly1 + poly2  poly4 = poly2 + poly1  print(poly3(-2), poly4(-2))  print(poly3(-1), poly4(-1))  print(poly3(0), poly4(0))  print(poly3(1), poly4(1))  print(poly3(2), poly4(2)) | 8 8  9 9  10 10  11 11  12 12 |